Der ROM-WIZARD ist eine Zusatzplatine, die in den ATARI 400/800 eingebaut wird.

Mit dieser Karte kann man Änderungen des Betriebssystems realisieren. Auf der Platine befinden sich insgesamt vier Sockel für EPROM- und ROM-Bausteine. In zwei der Fassungen steckt man die ROM-Bausteine des Original-Betriebssystems. Eine weitere Fassung ist für ein EPROM vom Typ 2764 (8K*8) vorgesehen, das ein neues Betriebssystem enthält. Mit einem Schalter ist es jetzt möglich, zwischen dem neuen und dem alten Betriebssystem umzuschalten. In den vierten Steckplatz wird ein 2532-EPROM eingesteckt. Es belegt den Bereich von $C000-$CFFF.

Wir liefern einen Monitor für diesen Bereich und ein geändertes Betriebssystem mit, das von jedem Programm aus den Sprung in diesen Monitor erlaubt. Es ist so möglich ein Programm einzuladen, zu unterbrechen und in den Monitor zu springen. Man kann sich das Programm nun Auflisten, den Speicher ganz oder teilweise auf Diskette schreiben (auch ohne DOS !!!), einzelne Speicherstellen ändern usw. Es gibt drei Möglichkeiten, ein Programm zu listen; als HEX-, als Charakter- und als Assembler-Liste. Außerdem bietet der Monitor noch viele weitere Hilfen zum Austesten von Programmen. (Programm stoppen, Prozessor-Register ändern, wieder starten; Einzelschritte ausführen usw.)

Eine weitere Änderung, die unser neues Betriebssystem enthält, ist ein neuer Drucker-Handler. Dadurch kann man nun mit allen Programmen unser kleines Drucker-interface (wird an 3. und 4. Joystick-Port ange-schlossen) benutzen.

Dem Programmierer sind bei Änderung des Betriebssystems keine Grenzen gesetzt. Es ist sogar denkbar, ein völlig neues Operating-System zu schreiben (z.B. XL-Operating-system). Dabei ist aber immer volle Kompatibilität zu allen Programmen gewährleistet, da man ja auf das alte Betriebssystem umschalten kann. Die beiden EPROMS werden dann abgeschaltet, so daß sich der Computer wie ein ATARI 400/800 ohne Zusatzkarte verhält.

Der Einbau ist recht einfach. Die Original-ROMS werden auf die Zusatzplatine umgesteckt und der ROM-WIZARD wird über ein Flachbandkabel mit der Grundplatine des ATARI 400 bzw. dem ROM-Modul des ATARI 800 verbunden. Dann muß noch ein Verbindungskabel angelötet werden.

Der ROM-WIZARD kostet bei uns mit genauer Beschreibung und Einbau-Anleitung DM 280,-.
Wer sich nicht zutraut, die Karte selbst einzubauen, kann sie für DM 20,- bei uns einbauen lassen.

Das kleine Druckerinterface (läuft dann mit allen Programmen ohne Zusatz-Software) kostet DM 90,-

Beschreibung des ROM-WIZARD-MONITORS

****

Wie startet man den MONITOR ?

Um den MONITOR starten zu können, ist das von uns mitgelieferte Betriebssystem etwas modifiziert worden. Bei Drücken der RESET-Taste wird geprüft, ob die OPTION-oder die SELECT-Taste gedrückt ist. Wenn ja, so wird ein Sprung in den MONITOR ausgeführt.

Man kann also wie gewohnt ein Programm einladen, und dann durch Drücken von SELECT und RESET in den MONITOR gelangen. Nun kann das Programm geändert, aufgelistet, neu gestartet werden usw.

Der ROM-WIZARD ist so eine komfortable Möglichkeit jedes einladbare Programm zu unterbrechen (an jeder beliebigen Stelle!) und es dann zu bearbeiten.

Dazu bietet er viele komfortable Befehle die im Folgenden einzeln erklärt werden.

Bei Einsprung in den MONITOR meldet er sich mit folgenden Zeilen:

```
          ROM-WIZARD-MONITOR

  PC    STATUS ACCUM X-REG Y-REG STACK
Cxxxx     xx     xx    xx    xx    xx
```

Die angegeben Werte für den Programmzähler (PC), das Statusregister (STATUS), den Akkumulator (ACCUM), das X-Register (X-REG), das Y-Register (Y-REG) und den Stack (STACK) sind die Werte, die die entsprechenden Register bei Abbruch des Programms hatten.

Sie werden vom MONITOR gesondert gespeichert und falls ein Rücksprung in das Programm ausgeführt wird, vorher in die Register zurückgeschrieben.

Um das Programm wieder zu starten, drückt man START und RETURN gleichzeitig.

Ein sehr nützliches Kommando ist das Hilfs-Kommando. Bei diesem Kommando gibt der Monitor eine Liste seiner Befehle aus. Um das Hilfs-Kommando auszuführen, schreibt man: ´? RETURN´

Danach erscheint auf dem Bildschirm:

```
REG     :C
DUMP    :D (adr adr)
TRACE   :E (byt)
JSR     :J adr
DR/LINK:L (drive )
PRINTER:P
READ    :R (sec  adr  )
SUCHEN :S adr byt byt ...
HEX-CHR:T
WRITE   :W (sec  adr  )
DISASS :X (adr adr)
PUSH    :+ byt byt ...
POP     :-
```

Es werden die Befehle und die eventuell benötigten Parameter angezeigt. Stehen die Parameter in Klammern so ist die Angabe nicht unbedingt notwendig. Der Monitor benutzt dann z.B. beim Listen von Speicherbereichen immer die nächste Adresse.

Hier noch einige Hinweise zur Anwendung der Kommandos:

- Alle numerischen Ein- bzw. Ausgaben werden hexadezimal berechnet. (außer beim Speicher-Dump)
- Die einzelnen Parameter in einer Kommando-Zeile müssen durch ein Leer-Zeichen oder Komma getrennt werden.
- Die Ausführung eines Kommandos wird gestoppt, wenn der MONITOR auf ein illegales Parameter trifft.
- Um die Ausführung eines Kommandos zu unterbrechen wird die START-Taste gedrückt (nicht bei allen Kommandos möglich)

Speicher-Dump: D (Start Adresse) (End Adresse)

Dieses Kommando wird benutzt um den Inhalt eines Speicherbereiches auf dem Bildschirm abzubilden. Die Angabe der Adressen ist optional. Wird nur eine Start-Adresse angegeben, wird der Inhalt von acht Speicherstellen, von der Start-Adresse an, ausgegeben. Wenn man eine Start- und End-Adresse angibt wird ein entsprechender Bereich ausgegeben. Wird das ´D´-Kommando ohne Adressen benutzt, nimmt der Monitor die zuletzt benutzte Adresse als gültig an.

HEX/Charakter-Verarbeitung: T

Normalerweise wird der Speicher als HEX-Listing ausgegeben. Mit dem ´T´-Kommand kann man aber auf Charakter-Ausgabe schalten.

Der Monitor meldet dem Benutzer nach Ausführung des Befehls, daß er auf den Charakter-Modus umgeschaltet hat:

CHAR

Um wieder auf HEX-Verarbeitung umzuschalten, führt man nocheinmal den ´T´-Befehl aus:

T ´RETURN´

Rückmeldung:

HEX

Änderung des Speichers: A Adresse Byte Byte ...

Bei der Ausgabe von Speicherinhalten, wird ein ´A´ gefolgt von der Adresse und den Daten (in HEX oder als CHAR) ausgebeben, je durch ein Leer-Zeichen getrennt (nur im HEX-Modus). Will man den Speicherinhalt ändern, so braucht man ihn nur auszulisten und kann nun direkt in der entsprechenden Zeile die Änderung vornehmen. (Mit ´RETURN´ abschließen)

Das ist dadurch möglich, daß bei der Ausgabe jeder Zeile ein ´A´ vorangestellt wird. Das ist nämlich für den Monitor ein neues Kommando zur Eingabe in den Speicher. Mit

A A000 1A 2B 3C

werden die Bytes 1A 2B 3C von der Adresse A000 an im Speicher abgelegt. Wenn vorher auf Charakter-Verarbeitung umgeschaltet worden ist, so werden die Zeichen hinter der Adresse als Charakter verarbeitet. Die einzelnen Buchstaben dürfen dann nicht durch Leer-Zeichen getrennt werden, da die Leer-Zeichen selbst als Charakter verarbeitet werden.

Suchen im Speicher: S Adresse Byte Byte ...

Um im Speicher nach einer Folge von HEX-Bytes bzw. Charaktern zu suchen, gibt es das ´S´-Kommando. Um es auszuführen, schreibt man ´S´ gefolgt von der Adresse in HEX, einem Leer-Zeichen und der Such-Sequenz als HEX-Bytes bzw. Charakter-Folge:

S 1000 1A 2B 3C

Der MONITOR durchsucht den Speicher von der Adresse 1000 an auf die Byte-Folge 1A 2B 3C. Um den ganzen Speicher zu durchsuchen, braucht er dabei nur wenige Sekunden. Wenn er die Sequenz findet, gibt er die Adresse, an der die Sequenz steht, aus. Steht die gleiche Folge mehrmals im Speicher, werden mehrere Adressen ausgebeben. Die Länge der Such-Sequenz kann beliebig groß sein (wird durch die Länge der Eingabe-Zeile von 114 Zeichen begrenzt).

Druckerausgabe: P

Das P-Kommando ist für alle anwendbar, die einen Drucker besitzen. Nach diesem Befehl wird alles auf den Bildschirm ausgegebene auch ausgedruckt. Negative Zeichen werden dabei nicht negativ gedruckt, Zeichen, die der Drucker nicht als Charakter versteht (Steuerzeichen), werden als ´-´ an den Drucker ausgegeben. Die Druckerausgabe ist besonders im Einzelschritt-Modus sinnvoll, wenn Programme nicht den Graphik 0 Bildschirm benutzen.

Das ´P´-Kommando wird sowohl zum Ein- als auch zum Ausschalten der Drucker-Ausgabe benutzt. Wird der Befehl einmal ausgeführt, wird die Drucker-Option ein-, bei nochmaliger Ausführung wieder ausgeschaltet.

Disketten Ein-/Ausgabe

Die Disketten-Bearbeitung wird unabhängig vom DOS durchgeführt. Man kann einzelne oder auch mehrere Sektoren lesen bzw. schreiben. Die Sektoren werden entweder sequentiell oder mit Hilfe der Links verarbeitet. Das heißt es lassen sich sogar DOS-Files lesen, ohne daß das DOS im Speicher steht!

LINK/SEQ, Laufwerk: L (Laufwerks Nr.)

Normalerweise arbeitet der MONITOR im sequentiellen Modus. Um auf Verarbeitung mit Links umzuschalten, gibt es das ´L´-Kommando. Es dient wie z.B. der ´T´-Befehl zum Hin- und Herschalten zwischen den beiden Modi. Damit der Benutzer weiß, in welchem Modus sich der MONITOR gerade befindet, wird er mit ´SEQ´ oder ´LINK´ informiert.

Wird mit dem Kommando ein Parameter angegeben, so interpretiert der Monitor dieses als Laufwerksnummer und führt dann alle Ein-/Ausgaben über das neugewählte Laufwerk durch.

Der ´L´-Befehl hat also zwei Bedeutungen. Ohne Parameter schaltet er zwischen den beiden Modi (SEQ/LINK) um, mit Parameter schaltet er die Laufwerksnummer um.

Sektoren lesen: R (Sektor Nr.) (Adresse) (Zahl d. S.)

Das Lesen von Sektoren wird mit den ´R´-Kommando durchgeführt. Um einen Sektor zu lesen, schreibt man ´R´ gefolgt von der Sektornummer. Der gewählte Sektor wird jetzt in den Bereich ab 6000 (HEX) gelesen. Will man den Sektor in einen anderen Speicherbereich schreiben, dann wird noch ein Leer-Zeichen und die entsprechende Adresse angehängt:

R1 1000

Der Sektor Nr.1 wird in den Speicher ab 1000 abgelegt. Soll mehr als einen Sektor gelesen werden, wird noch die Zahl der Sektoren angegeben:

R1 1000 20

Damit werden 20 (HEX,=32 dezimal) Sektoren von Nr.1 an in den Speicher ab 1000 abgelegt. Dabei wird automatisch auf den jeweils gewählten Modus (SEQ/LINK) geachtet.

Beim Lesen von Sektoren, wird für jeden Sektor die gerade gelesene Sektor-Nummer und die Speicheradresse, in die er abgelegt wird, angezeigt.

Wenn das ´R´-Kommando einmal ausgeführt wurde, brauchen die Parameter nicht jedesmal neu angegeben zu werden. Um den nächsten Sektor zu lesen, schreibt man einfach ´R´ und ´RETURN´.

WENN DIE DRUCKER-AUSGABE EINGESCHALTET IST, MÜSSEN DIE PARAMETER JEDESMAL ANGEGEBEN WERDEN, WEIL SICH DRUCKER UND DISKETTE DEN I/O-BLOCK TEILEN.

Ein DOS-File läd man am besten durch Angabe einer übergroßen Zahl von Sektoren ein. Der Monitor stoppt den Lesevorgang automatisch am Ende des Files und gibt ´EOF´ (END OF FILE) aus.

Ein kleiner Hinweiß: Die Directory einer Diskette liegt in den Sektoren 169-170 (HEX). Um die Directory einzulesen, wird die folgende Zeile eingegeben:

R169 6000 8

Im Charakter-Modus kann man sich jetzt die Directory ansehen. Die 5 Bytes, die vor jedem Namen stehen haben folgende Bedeutung:

1. Byte	2.+3. Byte	4.+5. Byte
File-Status	Zahl der Sektoren	Nr. des 1. Sektors

(Der File-Status enthält Angaben darüber, ob das File gelöscht ist usw.)

Sektoren schr.: W (Sektor Nr.) (Adresse) (Zahl d. S.)

Das Schreiben von Sektoren aus dem Speicher auf die Diskette geschieht analog zum Lesevorgang. Die Syntax für Sektornummer, Bufferadresse, Zahl der Sektoren ist die gleiche wie beim ´R´-Kommand:

W1 6000 10

Diese Zeile veranlasst den MONITOR 10 (=16 dezimal) Sektoren aus dem Speicher ab 6000 in die Sektoren Nr.1 bis Nr.16 zu schreiben.

Es ist jedoch nicht möglich, beim Schreiben im LINK-Modus zu arbeiten!

Weitere Möglichkeiten des MONITORS

Wir haben gesehen, daß es zwei Möglichkeiten gibt, den MONITOR zu starten. Nämlich OPTION-RESET und SELECT-RESET. Wird OPTION-RESET gedrückt, so wird ein normaler Warmstart und anschließend ein Unterprogramm-Sprung (JSR) in den Monitor ausgeführt. Wenn man START-RETURN drückt, wird der MONITOR verlassen und der Warmstart wird vollendet.

Eine effektivere Möglichkeit ist die SELECT-RESET Kombination. Dann wird nämlich kein Warmstart ausgeführt und auch der Stack wird nicht zerstört. Die Zero-Page bleibt unverändert, so daß der Programm-ablauf einfach mit START-RETURN wieder fortgeführt werden kann. (Probleme gibt es dabei, wenn das Programm nicht den Graphik 0 Bildschirm benutzt).
Außerdem kann der Ablauf des Programms besser zurückverfolgt werden (über den STACK), oder der momentanen Speicher mit dem ´W´-Kommando auf Diskette abgespeichert werden.

Um Programme gezielt an bestimmten Stellen zu unterbrechen, kann man entweder eine BRK-Instruktion oder ein JSR $C001 in das Programm einfügen.

CPU-Resgister anzeigen, ändern:

C (PC) (STATUS) (AKK) (X-REG) (Y-REG)

Wird das ´C´-Kommando ohne Parameter ausgeführt, so werden die Register-Inhalte bei Verlassen des Programms angezeigt. Will man die Register vor Rücksprung in das Programm ändern, so gibt man die neuen Inhalte als Parameter hinter dem Kommando an. Der STACK ist das einzige Register, das nicht mit diesem Befehl geändert werden kann. Zur STACK-Änderung stehen die folgenden Befehle zur verfügung:

STACK-OPERATIONEN: + Byte Byte ...
-

Mit ´+´ können ein oder mehrere Bytes im STACK abgelegt werden. Dabei wird der STACK nach jedem Byte um eins heruntergezählt. Das erste Byte hinter ´+´ wird auch als erstes in den STACK geschoben:

+ 01 FA

zuerst wird 01, dann FA in den STACK geschrieben

Mit dem '-'-Kommando wird immer ein Byte vom STACK geholt und auf dem Bildschirm angezeigt. Zum STACK wird dabei immer eine Eins dazugezählt.

Disassemblieren: X (Start Adresse) (End Adresse)

Der Speicher kann nicht nur als HEX- oder Charakter-Listing ausgegeben werden. Mit dem 'X'-Kommando kann auch ein Programm aus dem Speicher als Assembler-Liste ausgegeben werden. Die Parameter für Start- und End-Adresse werden genau so angegeben, wie beim 'D'-Befehl. Diese Funktion erleichtert das Ansehen von Programmen, da Programme als Assembler-Listing wesentlich leichter zu lesen und zu verstehen sind.

Einzelschritt/Trace: E (spez. Optionen/Zahl d. Schr.)

Das 'E'-Kommando ohne Parameter, veranlaßt den MONITOR, den nächsten Befehl auszuführen und danach wieder alle Register anzuzeigen. Man kann jetzt z.B. die Register Ändern und dann mit 'E' in dem Programm fortfahren. Will man mehr als nur einen Schritt ausführen, so gibt man die Zahl der Schritte hexadezimal als Parameter an. Die größte mögliche Zahl ist hier 1F (=31 dezimal). Gibt man also:

E 1A

ein, so werden 1A (=26 dezimal) Schritte ausgeführt. Die Register werden aber nach jedem Befehl angezeigt.

Die untersten 5 Bit des Parameter-Bytes, geben die Schrittzahl an, die obersten 3 haben spezielle Bedeutungen.

Wenn Bit Nr. 8 gesetzt ist, so wird die Zahl der Schritte nicht beachtet und das Programm läuft so lange, bis ein Abbruch durck Break, RESET oder ähnlich erfolgt. 'E 80' entspricht also dem 'T'-Kommando (Trace) des Atari-Assemblers.

Wenn Bit Nr. 7 gesetzt ist, so werden die Register nur auf den Drucker (wenn der Drucker mit 'P' eingeschaltet wurde), aber nicht auf den Bildschirm ausgegeben. Das ist sinnvoll, wenn das zu testende Programm nicht den Graphik 0 Bildschirm benutzt.

Durch Setzten des 6. Bits, werden die Register-Inhalte nur nach jedem 32. Befehl ausgeben.

Ein Vorteil, der Trace-Funktion des ROM-WIZARD-MONITORS, ist es, daß ein Sprung in das OS als eine einzige Instruktion behandelt wird. Das heißt, das die Trace-Ausgabe bis zum Rücksprung unterdrückt wird. Alle Adressen, die über $C000 liegen, werden als OS-Adressen angesehen.

Unterprogramm-Sprung: J Adresse

Dieses Kommando, ermöglicht es, eine Unterroutine anzuspringen und nach Abschluß (RTS) wieder in den MONITOR zurückzugelangen. Es kann natürlich auch benutzt werden, um OS-Routinen anzuspringen. Das ist z.B. zum Formatieren einer Diskette notwendig, da der MONITOR selbst keine Foramtierung durchführen kann. Das ´J´-Kommando selbst führt nicht den Sprung aus, sondern trifft nur die entsprechenden Vorbereitungen, um nach RTS in den MONITOR zurückzuspringen. Um das Unterprogramm zu starten, drückt man START-RETURN.

Beispiel zur Disketten-Formatierung:

Die Drucker-Option muß ausgeschaltet sein!
festlegen der benötigten Parameter:

A 301 01 21
J E453

START-RETURN

Nachdem ein Programm unterbrochen wurde, ist es teilweise nötig, eine neue Initialisierung vorzunehmen, bevor das Programm neu gestartet wird. Auch dazu kann der ´J´-Befehl verwendet werden. (Die Initialisierungs-Adresse, steht normalerweise in den Adressen 0A und 0B)

